Amendments to the Claims:

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-13. (Canceled)

14. (Currently Amended) The method as recited in claim 16 ~~of Claim 13~~ wherein the resolving of the miss exception event includes pausing the miss exception handler until the miss exception event is resolved.

15. (Canceled)

16. (Currently Amended) A method of ~~The method of Claim 13~~ handling translation lookaside buffer (TLB) miss exceptions in a memory management unit of a multi-processor computer system having a plurality of Central Processing Units (CPU's), the method comprising:

initiating an access instruction;

determining that a TLB miss exception event has occurred, wherein determining that a TLB miss exception event has occurred includes testing a context identifier for the affected virtual address to determine whether a TLB miss exception has occurred;

invoking a miss exception handler;

determining the nature of the TLB miss exception event;

resolving the miss exception event;

returning to initiating an access instruction;

wherein determining the nature of the TLB miss exception event includes testing the context identifier for the affected virtual address to determine whether the TLB miss exception event results from one of an unassigned context identifier, a translation storage buffer (TSB) resizing operation that affects the virtual address, and an unmapping of a shared translation table entry (TTE) that is associated with the virtual address wherein the TTE comprises a shared memory resource;

~~wherein determining the nature of the TLB miss exception event includes testing a context identifier for the affected virtual address to determine whether the TLB miss exception event results from one of an unassigned context identifier, a translation storage buffer (TSB) resizing operation that affects the virtual address, and an unmapping of a shared translation table entry (TTE) that is associated with the virtual address wherein the TTE comprises a shared memory resource;~~

wherein resolving the miss exception events includes resolving each type of miss exception event in accordance with specified miss exception resolution protocol for each type of miss exception event; and

wherein resolving the miss exception events includes resolving each type of miss exception event in accordance with specified miss exception resolution protocol for each type of miss exception event.

17. (Canceled)

18. (Previously Presented) The method of Claim 16 wherein said testing the context identifier determines that said unavailability results from a miss exception due to an unassigned context identifier, and

wherein resolving the unassigned context identifier miss exception event further includes the steps of:

assigning a context identifier value to a virtual address space that is to be associated with a process that contains the virtual address;

assigning selected portion of memory to a TSB that is to be associated with the virtual address space having the assigned context identifier;

updating at least one of secondary memory assets and the TLB with TSB and context identifier information; and

returning to initiating an access instruction.

19. (Previously Presented) The method of Claim 16 wherein said testing the context identifier determines that said unavailability results from a miss exception due to a situation wherein the affected virtual address maps to a shared memory resource, and

wherein resolving the miss exception event further includes the steps of:

determining if the shared memory resource is locked;

if the shared memory resource is not locked,

returning to initiating an access instruction; and

if the shared memory resource is locked,

pausing the exception handler until the shared memory resource becomes unlocked; and

returning to initiating an access instruction when the shared memory resource becomes unlocked.

20. (Original) The method of Claim 19 wherein said shared memory resource comprises a translation table entry (TTE).

21. (Previously Presented) The method of Claim 16 wherein testing the context identifier determines that said unavailability results from a miss exception due to a situation wherein the virtual address is associated with a TSB that is undergoing a resizing operation; and

wherein resolving the miss exception event further includes the steps of:

A) determining if the virtual address space of the TSB undergoing resizing is locked;

1) in a case wherein the virtual address space of the TSB undergoing resizing is locked,

i) pausing the exception handler until the virtual address space of the TSB undergoing resizing becomes unlocked;

ii) locking the virtual address space of the TSB undergoing resizing;

2) in a case wherein the virtual address space of the TSB undergoing resizing is unlocked,

i) locking the virtual address space of the TSB undergoing resizing;

B) once the virtual address space of the TSB undergoing resizing has been locked by one of 1)(ii) and 2(i), determining whether the TSB undergoing resizing has been assigned a specified location in memory;

1) if the TSB undergoing resizing has been assigned a specified location in memory,

i) releasing the lock on the virtual address space of the TSB undergoing resizing; and

ii) returning to initiating an access instruction;

2) if the TSB undergoing resizing has not been assigned a specified location in memory,

i) assigning a specified portion of memory to the TSB undergoing resizing;

ii) releasing the lock on the virtual address space of the TSB undergoing resizing; and

iii) returning to initiating an access instruction.

22-23. (Canceled)

24. (Currently Amended) A method ~~as in Claim 22~~ of accessing translation table entries (TTE's) in secondary memory assets of a memory management unit of a multi-processor computer system, the method comprising:

A) requesting that a translation be found for a selected virtual address having an associated context identifier wherein the translation is to be found in a secondary memory asset of a memory management unit;

B) testing the associated context identifier, prior to searching a secondary memory asset, to determine whether a TTE corresponding to the virtual address and the associated context identifier is available to have a memory access instruction executed upon it;

1) wherein if testing determines the TTE is available to have a memory access instruction executed upon it:

locating the TTE using the virtual address and context identifier;

accessing the TTE from the secondary memory asset;

updating a translation lookaside buffer (TLB) and the secondary memory asset as needed;

returning to initiating an access instruction;

2) wherein if testing determines the TTE is not available to have a memory access instruction executed upon it:

determining a source of unavailability;

resolving the unavailability;

returning to initiating an access instruction,

wherein the secondary memory assets include at least one translation storage buffer (TSB) and at least one page table, and

wherein resolving the unavailability includes:

determining the nature of the unavailability; and

where it is determined that the cause of the unavailability is a TSB resizing operation that affects the virtual address for which the translation is sought;

pausing until the TSB resizing operation is completed; and

returning to initiating an access instruction.

25. (Currently Amended) A method as in Claim [[22]] 24 wherein the secondary memory assets include at least one translation storage buffer (TSB) and at least one page table, and

wherein resolving the unavailability includes:

determining the nature of the unavailability; and

where it is determined that the cause of the unavailability is a demapping operation for a TTE that is shared by the virtual address for which the translation is sought,

pausing until the TTE demapping operation is completed; and

returning to initiating an access instruction.

26. (Previously Presented) A method of accomplishing memory management of miss exceptions in a memory management unit of a multi-processor computer system, the method comprising;

determining that a miss exception event has occurred, wherein the miss exception event concerns one of: an unassigned context identifier event, a memory access event changing a shared memory resource, and a translation storage buffer (TSB) resizing event;

resolving the miss exception event in accordance with a miss event resolution protocol suitable for resolving the received miss exception event; and

wherein said determining determines that the miss exception event comprises an instruction to change a translation table entry (TTE) that is shared by more than one virtual address space wherein each virtual address space has an associated context identifier; and

wherein resolving the miss exception event in accordance with a miss event resolution protocol comprises:

identifying virtual address spaces that share the same TTE;

activating a lock for each virtual address space that shares the same TTE to prevent other processes from accessing the TTE while the lock is activated;

changing the corresponding context identifier for each virtual address space that shares the same TTE thereby making the associated TTE unavailable to have memory access instructions performed thereon;

performing the changes on the TTE;

releasing the locks on each locked virtual address space; and

freeing the corresponding context identifiers for each affected virtual address space.

27. (Canceled)

28. (Original) The method of Claim 26 wherein said determining determines that the miss exception event comprises a miss exception event comprises an instruction to resize a translation storage buffer (TSB); and

wherein resolving the miss exception event in accordance with a miss event resolution protocol comprises:

activating a lock for the TSB to prevent other processes from accessing entries in the TSB while the lock is activated;

changing the corresponding context identifier to indicate that the TSB virtual address space is unavailable to have memory access instructions performed thereon;

resizing the TSB;

changing the corresponding context identifier back to its original configuration; and

releasing the lock on the TSB.

29. (Original) The method of Claim 26 wherein said determining determines that the miss exception event comprises an instruction concerning an unassigned context identifier; and

wherein resolving the miss exception event in accordance with a miss event resolution protocol comprises:

assigning a context identifier for a virtual address space associated with a TSB; and

assigning a portion of memory to the TSB.

30. (New) A computer readable medium including computing program code for handling translation lookaside buffer (TLB) miss exceptions in a memory management unit of a multi-processor computer system having a plurality of Central Processing Units (CPU's), wherein said computer readable medium comprising:

computer program code for initiating an access instruction;

computer program code for determining that a TLB miss exception event has occurred, wherein determining that a TLB miss exception event has occurred includes testing a context identifier for the affected virtual address to determine whether a TLB miss exception has occurred;

computer program code for invoking a miss exception handler;

computer program code for determining the nature of the TLB miss exception event;

computer program code for resolving the miss exception event;

computer program code for returning to initiating an access instruction;

wherein determining the nature of the TLB miss exception event includes testing the context identifier for the affected virtual address to determine whether the TLB miss exception event results from one of an unassigned context identifier, a translation storage buffer (TSB) resizing operation that affects the virtual address, and an unmapping of a shared translation table entry (TTE) that is associated with the virtual address wherein the TTE comprises a shared memory resource;

wherein resolving the miss exception events includes resolving each type of miss exception event in accordance with specified miss exception resolution protocol for each type of miss exception event; and

wherein resolving the miss exception events includes resolving each type of miss exception event in accordance with specified miss exception resolution protocol for each type of miss exception event.

31. (New) A computer readable medium as recited in claim 30, wherein resolving the unassigned context identifier miss exception event further includes the steps of:

assigning a context identifier value to a virtual address space that is to be associated with a process that contains the virtual address;

assigning selected portion of memory to a TSB that is to be associated with the virtual address space having the assigned context identifier;

updating at least one of secondary memory assets and the TLB with TSB and context identifier information; and

returning to initiating an access instruction.

32. (New) A computer readable medium as recited in claim 30, wherein said shared memory resource comprises a translation table entry (TTE).

33. (New) A computer readable medium as recited in claim 30, wherein resolving the miss exception event further includes the steps of:

determining if the shared memory resource is locked;

if the shared memory resource is not locked,

returning to initiating an access instruction; and

if the shared memory resource is locked,

pausing the exception handler until the shared memory resource becomes unlocked; and

returning to initiating an access instruction when the shared memory resource becomes unlocked.

34. (New) A computer readable medium as recited in claim 30, wherein resolving the miss exception event further includes the steps of:

A) determining if the virtual address space of the TSB undergoing resizing is locked;

1) in a case wherein the virtual address space of the TSB undergoing resizing is locked,

i) pausing the exception handler until the virtual address space of the TSB undergoing resizing becomes unlocked;

ii) locking the virtual address space of the TSB undergoing resizing;

2) in a case wherein the virtual address space of the TSB undergoing resizing is unlocked,

i) locking the virtual address space of the TSB undergoing resizing;

B) once the virtual address space of the TSB undergoing resizing has been locked by one of 1)(ii) and 2(i), determining whether the TSB undergoing resizing has been assigned a specified location in memory;

1) if the TSB undergoing resizing has been assigned a specified location in memory,

i) releasing the lock on the virtual address space of the TSB undergoing resizing; and

ii) returning to initiating an access instruction;

2) if the TSB undergoing resizing has not been assigned a specified location in memory,

i) assigning a specified portion of memory to the TSB undergoing resizing;

ii) releasing the lock on the virtual address space of the TSB undergoing resizing; and

iii) returning to initiating an access instruction.

35. (New) A multi-processor computing system having a plurality of Central processing Unites (CPUs), wherein said multi-processor is operable for handling translation lookside buffer (TLB) miss exceptions in a memory management unit of said multi-processor computer system, and wherein said multi-processor computing system is further operable to:

initiating an access instruction;

determining that a TLB miss exception event has occurred, wherein determining that a TLB miss exception event has occurred includes testing a context identifier for the affected virtual address to determine whether a TLB miss exception has occurred;

invoking a miss exception handler;

determining the nature of the TLB miss exception event;

resolving the miss exception event; and

returning to initiating an access instruction.

wherein determining the nature of the TLB miss exception event includes testing the context identifier for the affected virtual address to determine whether the TLB miss exception event results from one of an unassigned context identifier, a translation storage buffer (TSB) resizing operation that affects the virtual address, and an unmapping of a shared translation table entry (TTE) that is associated with the virtual address wherein the TTE comprises a shared memory resource;

wherein resolving the miss exception events includes resolving each type of miss exception event in accordance with specified miss exception resolution protocol for each type of miss exception event; and

wherein resolving the miss exception events includes resolving each type of miss exception event in accordance with specified miss exception resolution protocol for each type of miss exception event.

36. (New) A multi-processor computing system as recited in claim 35, wherein said testing the context identifier determines that said unavailability results from a miss exception due to an unassigned context identifier, and

wherein resolving the unassigned context identifier miss exception event further includes the steps of:

assigning a context identifier value to a virtual address space that is to be associated with a process that contains the virtual address;

assigning selected portion of memory to a TSB that is to be associated with the virtual address space having the assigned context identifier;

updating at least one of secondary memory assets and the TLB with TSB and context identifier information; and

returning to initiating an access instruction.

37. (New) A multi-processor computing system as recited in claim 35, wherein said testing the context identifier determines that said unavailability results from a miss exception due to a situation wherein the affected virtual address maps to a shared memory resource, and

wherein resolving the miss exception event further includes the steps of:

determining if the shared memory resource is locked;

if the shared memory resource is not locked,

returning to initiating an access instruction; and if the shared memory resource is locked,

pausing the exception handler until the shared memory resource becomes unlocked; and

returning to initiating an access instruction when the shared memory resource becomes unlocked.

38. (New) A multi-processor computing system of accomplishing memory management of miss exceptions in a memory management unit of a multi-processor computer system, the method comprising;

determining that a miss exception event has occurred, wherein the miss exception event concerns one of: an unassigned context identifier event, a memory access event changing a shared memory resource, and a translation storage buffer (TSB) resizing event;

resolving the miss exception event in accordance with a miss event resolution protocol suitable for resolving the received miss exception event; and

wherein said determining determines that the miss exception event comprises an instruction to change a translation table entry (TTE) that is shared by more than one virtual address space wherein each virtual address space has an associated context identifier; and

wherein resolving the miss exception event in accordance with a miss event resolution protocol comprises:

identifying virtual address spaces that share the same TTE;

activating a lock for each virtual address space that shares the same TTE to prevent other processes from accessing the TTE while the lock is activated;

changing the corresponding context identifier for each virtual address space that shares the same TTE thereby making the associated TTE unavailable to have memory access instructions performed thereon;

performing the changes on the TTE;

releasing the locks on each locked virtual address space; and

freeing the corresponding context identifiers for each affected virtual address space.

39. (New) A multi-processor computing system as recited in claim 35, wherein testing the context identifier determines that said unavailability results from a miss exception

due to a situation wherein the virtual address is associated with a TSB that is undergoing a resizing operation; and

wherein resolving the miss exception event further includes the steps of:

A) determining if the virtual address space of the TSB undergoing resizing is locked;

1) in a case wherein the virtual address space of the TSB undergoing resizing is locked,

i) pausing the exception handler until the virtual address space of the TSB undergoing resizing becomes unlocked;

ii) locking the virtual address space of the TSB undergoing resizing;

2) in a case wherein the virtual address space of the TSB undergoing resizing is unlocked,

i) locking the virtual address space of the TSB undergoing resizing;

B) once the virtual address space of the TSB undergoing resizing has been locked by one of 1)(ii) and 2(i), determining whether the TSB undergoing resizing has been assigned a specified location in memory;

1) if the TSB undergoing resizing has been assigned a specified location in memory,

i) releasing the lock on the virtual address space of the TSB undergoing resizing; and

ii) returning to initiating an access instruction;

2) if the TSB undergoing resizing has not been assigned a specified location in memory,

i) assigning a specified portion of memory to the TSB undergoing resizing;

ii) releasing the lock on the virtual address space of the TSB undergoing resizing; and

iii) returning to initiating an access instruction.